Assessment of the Torus Algorithm for Global Optimization

Richard Denny

Biophysics Section, Blackett Laboratory, Imperial College, London SW7 2BZ
&
CCLRC Daresbury Laboratory, Warrington, Cheshire WA4 4AD

Methods for global optimization of a parameter set as judged against some objective function are particularly relevant to the problem of optimizing a model against a set of diffraction data. There are many methods available for refining a model once in the neighbourhood of the desired minimum which take only downhill steps. However, methods which can escape the neighbourhood of a local minimum and locate the best solution are less common, an example being a simple grid search. This is usually too expensive in terms of the number of function evaluations required for problems involving more than a few variables. Other methods include simulated annealing and the so-called genetic algorithms.

The Torus algorithm (Rabinowitz, 1995) is a stochastic method which uses only function evaluations, no derivative calculations are required. The algorithm employs random shifting of variables within two different schemes for limiting the maximum range of those shifts. Initially, the ranges of the variables are defined by lower and upper bounds supplied by the user. Thereafter, in the controlling function, the variables are sequentially bumped around the set which currently gives the lowest function value, the range of the bumps decreasing exponentially with a low decay constant (slow cooling). The controlling function calls two routines for minimizing the objective function about the bumped variable set, one which applies random shifts to all variables and one which applies shifts to only one variable at a time. Both these routines reduce the maximum range of the shifts logarithmically (rapid cooling) from the ranges modified by the controlling function. The shifts are also constrained by a cutoff value for each variable supplied by the user. In the controlling function, a multiple of the cutoff value is used as the minimum size of the range. In the minimization routines, the cutoffs provide the minimum magnitudes of the random shifts. Therefore, the cutoffs can be thought of as defining minimum starting and finishing temperatures for the minimization routines. The implementation of this method investigated here was written in C from pseudocode published in Rabinowitz (1995).

A simple function with many local minima was used to investigate the efficacy of the Torus algorithm,

$$f_{n,k}(x) = \sum_{i=1}^{n} x_i^2 + k\left(1 - \prod_{i=1}^{n} \cos x_i\right)$$

where n is the number of dimensions and k is a weighting factor between the quadratic term which defines the global minimum and the periodic term which introduces the local minima. It is clear by inspection of the function that the global minimum has a value of 0 and is situated at $x_i=0$ for $i=1,n$. The family of next lowest minima have a value of

	TORUS			REPLEX	
n , k	No. of global minima	Av. global minimum	Av. no. of function calls	Av. global minimum	Av. no. of function calls
2 , 400	15	26.99	8054	5.47×10^{-9}	120
2 , 200	32	19.14	8104	5.01×10^{-9}	114
2 , 100	49	9.35	8585	4.93×10^{-9}	100
10 , 400	14	0.013	143082	1.79×10^{-6}	592
10 , 200	26	7.34	149014	3.52×10^{-6}	607
10 , 100	47	5.52	156182	1.14×10^{-4}	591

Table 1 Each combination of n and k was tested 100 times. The averages of the number of functions calls were calculated over all trials, not just those in which the neighbourhood of the global minimum was located.

approximately $2\pi^2$ and occur when two of the $x_i=\pm\pi$ and the rest are equal to zero. The starting set for each trial consisted of $x_1=12\pi$ and $x_i=0$ for $i=2,n$ with the range for each variable being $-50 \leq x_i \leq 50$. The cutoff value applied for each dimension was $\Delta x=0.01$. A representation of the function is shown in figure 1. Control parameters for the algorithm were left at their default values.

Each test consisted of 100 trials of the algorithm for a particular n and k. In order to clearly identify the neighbourhood of the local minimum into which the result of the Torus algorithm fell, each application was followed by refinement by a modification (REPLEX) of the downhill simplex method (Nelder & Mead, 1965), in which the variable set is partitioned to form groups with similar shifts. The results of these tests are summarized in table 1. The results of the trials for $n=2$ are shown graphically in figure 2.

As can be seen from table 1, the Torus algorithm requires far more function evaluations than the downhill simplex method to reach convergence. However, it has the great advantage that in a fair proportion of the trials, it successfully locates the neighbourhood of the global minimum, whereas, from the same starting point, the downhill simplex method would barely move. The downhill simplex method comes into its own when used to refine solutions suggested by the Torus algorithm. One can also compare the efficiency of the algorithm to a grid search in increasing the dimension of the problem. The scaling of the number of function evaluations for $n=2$ to $n=10$ is approximately 18 with little or no loss of efficiency in finding the global minimum. The number of function evaluations required by a grid search for $n=10$, assuming the same number of evaluations as the Torus algorithm for $n=2$ (corresponding to a step size of 1.13 in each dimension), is approximately 3.3×10^{19}, a scaling of 4.1×10^{15}. Even though the Torus algorithm cannot guarantee to locate the global minimum, it seems to offer a considerable advantage over a grid search for a problem of more than 1 or 2 dimensions. As with any non-trivial method of finding a global minimum, the algorithm has a number of control parameters which can be tuned to suit the problem at hand. This tuning is in itself a tricky optimization problem which has not been explored here.

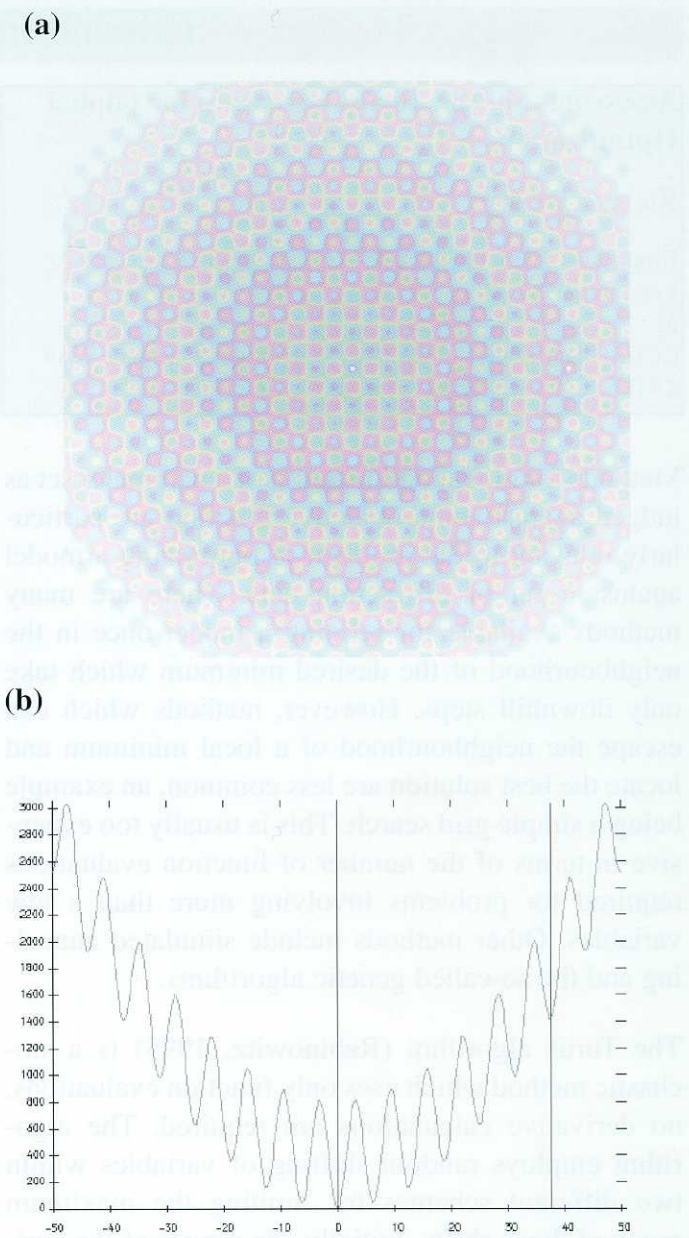

Figure 1 (a) shows a colour-contour representation of the test function for $n = 2$, $k = 400$, the limits in both dimensions being -50 to 50. The central global minimum is marked with a white dot as is the starting point to the right of the centre. (b) shows a slice through the centre of the surface with $x_2 = 0$. The central global minimum and the starting point are marked with vertical lines.

References

Rabinowitz F.M. 1995. Algorithm 744: A stochastic algorithm for global minimization with constraints. *ACM Trans. Math. Softw.*, **21**, 194-213.

Nelder, J.A. and Mead, R. 1965. A simplex method for function minimization. *Comput. J.*, **7**, 308-313.

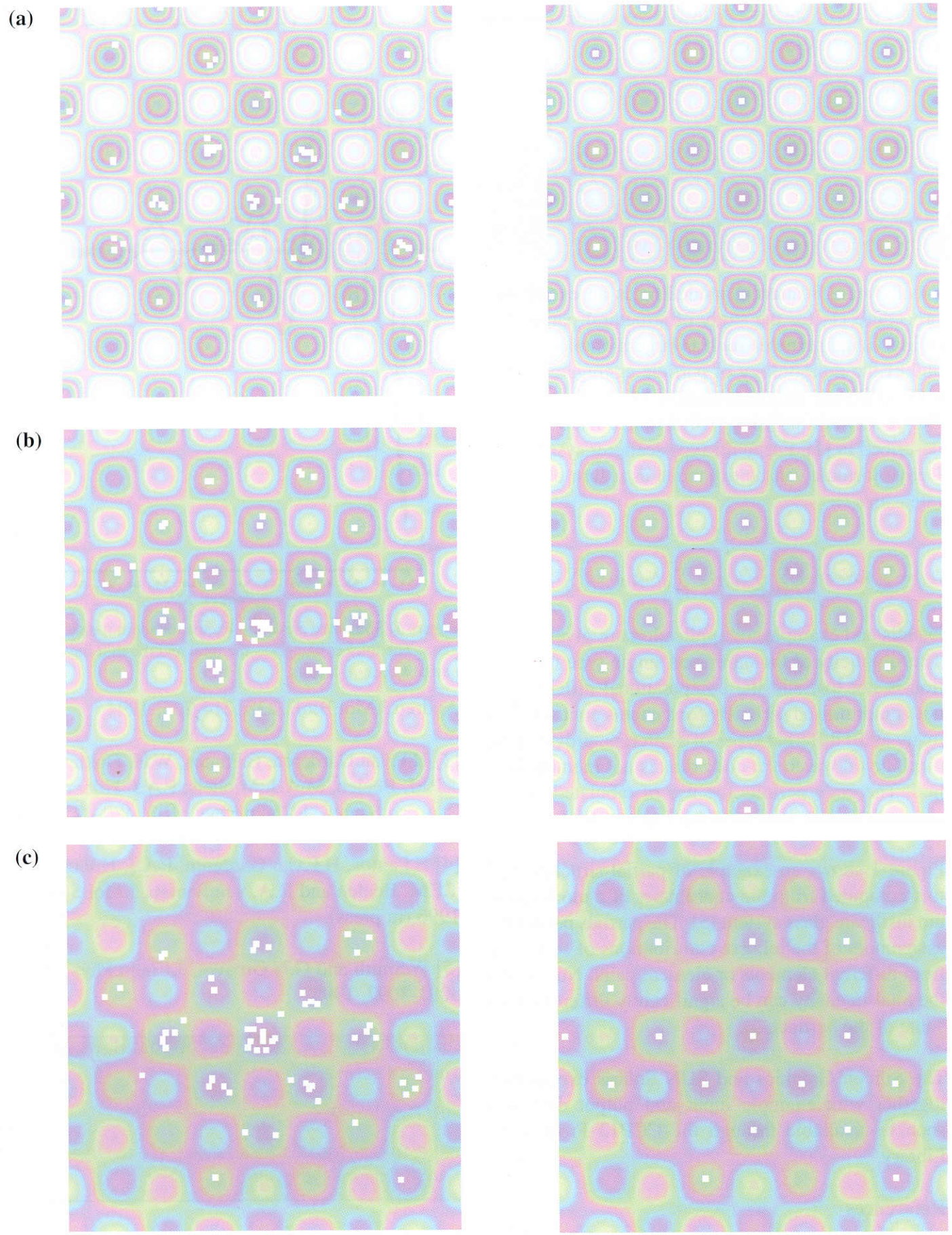

Figure 2 This shows the results of the trials for $n = 2$. The central portion of the function surface is shown ($-4\pi < x_1, x_2 < 4\pi$) with the resulting coordinates of the Torus algorithm on the left and the subsequent downhill simplex refinement on the right. (a) $k = 400$ (b) $k = 200$ (c) $k = 100$